REMARKS

This paper is responsive to the Office Action dated November 23, 2007. All rejections and objections of the Examiner are respectfully traversed. Reconsideration and further examination are respectfully requested.

At paragraphs 2-3 of the Office Action, the Examiner rejected claim 19 for non-statutory subject matter. Claim 19 has been canceled herein.

At paragraphs 4-5 of the Office Action, the Examiner rejected claims 1-23 for indefiniteness under 35 U.S.C. 112, second paragraph. Amendments to the claims herein are respectfully believed to meet all requirements of the Examiner in this regard.

At paragraphs 6-28, the Examiner rejected claims 1-23 for obviousness under 35 U.S.C. 103, citing the combination of U.S. patent 6691067 of Ding et al. ("Ding"), and U.S. patent 6330588 of Freeman ("Freeman"). Applicant respectfully traverses these rejections.

Ding discloses an enterprise management system that includes statistical recreation of system resource usage for monitoring, prediction, and performance workload characterization. In Ding, an agent computer system includes agent software for collecting data relating to one or more metrics, i.e., measurements of system resources. Metric data is continually collected by the agent software of Ding over the course of a measurement interval, regularly placed into a registry of metrics, and then periodically sampled from the registry indirectly.

Freeman discloses a system for verification of software agents and their activities in a distributed computing environment including an origin resource, a destination resource and a trusted resource. The origin resource of Freeman is associated with a software agent. The destination resource of Freeman is expected to advance the agent in the performance of an entrusted task, and the trusted resource is associated with the software agent in that the trusted resource functions to provide verification of the software agent and its activities. The trusted resource of Freeman supports one or more selected operations such as receiving/forwarding of software agents and other operations.

Nowhere in the combination of Ding and Freeman is there disclosed or suggested any method for monitoring system processor usage time of a software agent operating in a computer system, said method comprising the steps of:

> identifying said agent by associating an agent identifier therewith;
>
> initiating, responsive to said identifying of said agent, an agent lifetime timer for measuring an operating interval of said agent;
>
> *determining said operating interval using said lifetime timer by identifying a start time and a completion time of said agent and computing an elapsed time as the difference between said starting time and said completion time for said agent*; and
>
> storing said operating interval and said agent identifier in a computer-readable memory. (emphasis added)

as in the present independent claim 1. In contrast, Ding is concerned with using a software agent *to monitor an agent computer system*. For example, the agent 302 in Ding operates to report data regarding an agent node 300 to one or more monitors (e.g. 402). Ding states as follows beginning at line 21 in column 7:

> The data collectors 304 collect data from various processes and subsystems of the agent node 300. The Agent 302 sends real-time data to the UDR 210a, which is a database of historical data in a particular data format.

The above excerpt illustrates how the agents in Ding assist in monitoring the performance of a computer system node. In clear contrast, the present invention as set forth in independent claim 1 operates to *monitor resource use of a software agent by determining an operating interval using by identifying a start time and a completion time of the agent, and computing an elapsed time as the difference between the starting time and the completion time for the agent.* Combining the teachings of Freeman with Ding fails to remedy this shortcoming of the teachings of Ding, as Freeman is concerned with *verification of agents* and their activities, and not monitoring resource use of a software agent, as in the present independent claim 1.

Independent claim 13 also stands rejected based on the combination of Ding and Freeman. Independent claim 13 sets forth a method for monitoring system processor time usage of a software agent created by a thread associated therewith, said thread having a thread lifetime and said agent having an agent lifetime, said method comprising the steps of:

> associating an agent identifier with said agent;
>
> initiating, responsive to said associating said agent identifier with said agent, an agent lifetime timer for monitoring said agent lifetime;
>
> *determining system processor resource allocations of said agent, by identifying a start time and a completion time of said agent and computing said agent lifetime as the difference between said starting time and said completion time for said agent*, said resource allocations defining a footprint for said agent comprising:
>
> an amount of system processor resources utilized by said thread during said thread lifetime; and
>
> an amount of system processor resources utilized by said agent during said agent lifetime;
>
> associating said footprint with said agent identifier;
>
> storing said footprint and said agent identifier in a computer-readable memory;
>
> comparing said footprint of said agent to a plurality of footprints associated with a like plurality of other software agents;
>
> ranking said footprint of said agent against said plurality of footprints; and
>
> displaying those of said agent footprint and said plurality of footprints exceeding a predefined threshold. (emphasis added)

For reasons that should be clear from the discussion above, the combination of Ding and Freeman also does not disclose or suggest monitoring resource use of a software agent by *determining system processor resource allocations of the agent, by identifying a start time and a completion time of the agent and computing the difference between said starting time and said completion time for said agent,* as in the present independent claim 13.

Independent claim 20 also stands rejected based on the combination of Ding and Freeman. Claim 20 sets forth a method for tracking system processor time of a target agent operatively associated with a hypertext transport protocol process operating on a computer system and running a plurality of threads, said target agent further creating at least one of said plurality of threads, said method comprising:

> creating a computer-readable hash table in a memory operatively associated with said computer system;
>
> initiating an agent tracking function in machine-executable code in said computer system;
>
> identifying members of said plurality of threads by associating a thread identifier with each member of said plurality of threads producing a like plurality of identified threads;
>
> *identifying those of said plurality of identified threads created by said target agent to produce an identified thread set*;
>
> determining an amount of said system processor time utilized by said identified thread set; and
>
> storing said system processor time for said identified thread set in said hash table, thereby tracking said system processor time for said target agent. (emphasis added)

Neither Ding nor Freeman includes any teaching regarding identifying threads created by a target agent in order to track system processor time for a target agent, as in the present independent claim 20. Accordingly, the combination of Ding and Freeman does not disclose or suggest any system or method that includes indentifying those of a plurality of identified threads created by a target agent to produce an identified thread set, as in claim 20.

For the above reasons, Applicant respectfully urges that the combination of Ding and Freeman does not disclose or suggest all the features of the present independent claims 1, 13 and 20. The combination of Ding and Freeman therefore does not support a *prima facie* case of obviousness under 35 U.S.C. 103 with regard to claims 1, 13 and 20. As to the remaining claims, they each depend from claims 1, 13 and 20, and are respectfully believed to be patentable over the combination of Ding and Freeman for at least the same reasons.

Applicant has cancelled some claims from further consideration in this application, and have amended the independent claims. Applicant is not conceding in this application that the cancelled or unamended claims are not patentable over the art cited by the Examiner, as the present claim amendments and cancellations are only for facilitating expeditious prosecution of allowable subject matter. Applicant respectfully reserves the right to pursue the cancelled or unamended claims in one or more continuations and/or divisional patent applications.

Applicant has made a diligent effort to place the claims in condition for allowance. However, should there remain unresolved issues that require adverse action, it is respectfully requested that the Examiner telephone Applicant's Attorney at the number listed below so that such issues may be resolved as expeditiously as possible.

For these reasons, and in view of the above amendments, this application is now considered to be in condition for allowance and such action is earnestly solicited.

Respectfully Submitted,

February 20, 2008
Date

/David Dagg/
David Dagg, Reg. No. 37,809
Attorney/Agent for Applicant(s)
44 Chapin Road
Newton, MA 02459
(617) 630-1131

Docket No. 260-078